Text

Description automatically generated

STATISTICS FOR HIGH DIMENSIONAL DATA

**PROJECT REPORT**

**Classification of cardiac arrhythmia**

Mously DIAW – M2 IMSD

**Academic year 2020-2021**

Table of contents

[List of figures 3](#_Toc56874048)

[List of tables 3](#_Toc56874049)

[1. Introduction 4](#_Toc56874050)

[2. Data preprocessing and exploratory data analysis 4](#_Toc56874051)

[3. Data splitting and model strategy 4](#_Toc56874052)

[4. Results 5](#_Toc56874053)

[5. Discussion 6](#_Toc56874054)

[6. Conclusion 6](#_Toc56874055)

[7. Annexes 6](#_Toc56874056)

# List of figures

**No table of figures entries found.**

# List of tables

**No table of figures entries found.**

# Introduction

Overview. **Heart** rhythm problems (**heart arrhythmias**) occur when the electrical impulses that coordinate your heartbeats don't work properly, causing your **heart** to beat too fast, too slow or irregularly.

The cardiac arrhythmia is characterized by ir- regular rhythm of heartbeat which could be either too slow (\60 beats/min) or too fast ([100 beats/min) and can happen at any age.

Diagnosis of a heart arrhythmia involves measuring the heart activity for irregular heart beat using Electrocardiogram (ECG) and then analysing the recorded data. These parameters coupled with patient information can then be used by doctors to identify arrhythmia and its category.

Some challenges in identifying arrhythmia are:

Many of the current algorithms are rule based implementation but the cardio log’s classification is different and better.

Impossible for a doctor to identify minute steeps and irregularities due to the high number of parameters (i.e. > 270) involved.

**90% of clinical alarms in intensive care units might be false**. This high percentage negatively impacts both patients and clinical staff. The alarm overload might also lead to desensitization and could result in true alarms being ignored.

Hence we aim to create a classification model which will draw conclusions from the cardio log’s data as a gold standard and will distinguish between the presence and absence of cardiac arrhythmia and to classify it in one of the 16 groups from ECG data of new patients. This will help in achieving the below.

Reducing false alarms which in turn helps clinical staff to focus on the attention required areas.

Accurate detection to expedite patient’s treatment.

This database contains 279 attributes, 206 of which are linear valued and the rest are nominal.   
  
Concerning the study of H. Altay Guvenir: "The aim is to distinguish between the presence and absence of cardiac arrhythmia and to classify it in one of the 16 groups. Class 01 refers to 'normal' ECG classes 02 to 15 refers to different classes of arrhythmia and class 16 refers to the rest of unclassified ones. For the time being, there exists a computer program that makes such a classification. However there are differences between the cardiolog's and the programs classification. Taking the cardiolog's as a gold standard we aim to minimise this difference by means of machine learning tools.

1 Age: Age in years , linear   
2 Sex: Sex (0 = male; 1 = female) , nominal   
3 Height: Height in centimeters , linear   
4 Weight: Weight in kilograms , linear   
5 QRS duration: Average of QRS duration in msec., linear   
6 P-R interval: Average duration between onset of P and Q waves in msec., linear   
7 Q-T interval: Average duration between onset of Q and offset of T waves in msec., linear   
8 T interval: Average duration of T wave in msec., linear   
9 P interval: Average duration of P wave in msec., linear   
Vector angles in degrees on front plane of:, linear   
10 QRS   
11 T   
12 P   
13 QRST   
14 J   
  
15 Heart rate: Number of heart beats per minute ,linear

The goal of this experiment is to predict the class

due to class imbalance class attribute transformed into binary variable.

# Data preprocessing and exploratory data analysis

Describe data , common predictors

Outliers: z score>3 replaced by median => 2 values height

Missing values

Columns with missing values and percentage:

T 1.77 %

P 4.87 %

QRST 0.22 %

J 83.19 %

heart\_rate 0.22 %

drop j , knn imputation for the rest

drop columns with Near-zero variance (one single value or ratio two most frequent >20)

Drop highly correlated variables (corr>0.9)

FINAL dataset: 143 predictors, all numerical except sex and….

Describe..//scatter plot

Prior to modeling, one would expect that the main drivers of the execution time would be

As an initial investigation, Fig. shows the relationship between the classes using a mosaic plot.

Chart, box and whisker chart

Description automatically generated

# Data splitting and model strategy

There are 4331 samples available; 80% will be used for training the algorithms while the remainder will be used to evaluate the final candidate models. The data were split using stratified random sampling to preserve the class distribution outcome, Five repeats of 10-fold cross–validation were used to tune the models.

Rather than creating models that maximize the overall accuracy or Kappa statistics, f1 score function is used to

A series of models were fit to the training set. The tuning parameter com- bination associated with the smallest average cost value was chosen for the final model and used in conjunction with the entire training set. The following models were investigated:

Linear discriminant analysis: This model was created using the standard set of equations as well as with the penalized version that conducts fea- ture selection during model training. Predictor subset sizes ranging from 2 to 112 were investigated in conjunction with several values of the ridge penalty: 0, 0.01, 0.1, 1 and 10.

Partial least squares discriminant analysis: The PLS model was fit with the number of components ranging from 1 to 91.

Neural networks: Models were fit with hidden units ranging from 1 to 19 and 5 weight decay values: 0, 0.001, 0.01, 0.1, and 0.5.

Flexible discriminant analysis: First-degree MARS hinge functions were used and the number of retained terms was varied from 2 to 23.

Support vector machines (SVMs): Two different models were fit with the radial basis function. One using equal weights per class and another where the moderate jobs were given a fivefold weight and long jobs were up- weighted tenfold. In each case, the analytical calculations for estimating the RBF kernel function were used in conjunction with cost values ranging from 2−2 to 212 in the log scale.

Single CART trees: Similarly, the CART models were fit with equal costs per class and another where the costs mimic those in Table 17.2. In each case, the model was tuned over 20 values of the complexity parameter.

Bagged CART trees: These models used 50 bagged CART trees with and without incorporating the cost structure.

Random forests: The model used 2,000 trees in the forest and was tuned over 6 values of the tuning parameter.

# Results

The model results are shown in Fig. 17.5 where box plots of the resampling estimates of the mean cost value are shown. The linear models, such as LDA and PLS, did not do well here. Feature selection did not help the linear discriminant model, but this may be due to that model’s inability to han- dle nonlinear class boundaries. FDA also showed poor performance in terms of cost.

There is a cluster of models with average costs that are likely to be equivalent, mostly SVMs and the various tree ensemble methods. Using costs/weights had significant positive effects on the single CART tree and SVMs. Figure 17.6 shows the resampling profiles for these two models in

terms of their estimates of the cost, overall accuracy and Kappa statistic. The CART model results show that using the cost has a negative effect on accuracy and Kappa but naturally improved the cost estimates. No matter the metric, the tuning process would have picked the same CART model for final training

Trees clearly did well for these data, as did support vector machines and neural networks. Is there much of a difference between the top models? One approach to examining these results is to look at a confusion matrix generated across the resamples. Recall that for resampling, there were 50 hold–out sets that contained, on average, about 347 jobs. For each one of these hold–out sets, a confusion matrix was calculated and the average con- fusion matrix was calculated by averaging the cell frequencies.

Table 17.3 shows two such tables for the random forests model and the cost-sensitive CART model. For random forest, the average number of long jobs that were misclassified as very fast was 0.2 while the same value for the classification tree was 0.24. The CART tree shows very poor accuracy for the fast jobs compared to the random forest model. However, the opposite is true for moderately long jobs; random forest misclassified 72.56% of those jobs (on average), compared to 65.52% for the single tree. For long jobs, the single tree has a higher error rate than the ensemble method. How do these two models compare using the test set? The test set cost for random forest was 0.316 while the single classification trees had a average cost of 0.37. Table 17.4 shows the confusion matrices for the two models. The trends in the test set are very similar to the resampled estimates.

long jobs. In summary, the overall differences between the 2 models s not large.

# Discussion

# Conclusion

# Annexes